UNIVAC 1832 COMPUTER
REPERTOIRE OF INSTRUCTIONS

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
01 0	OR	Inclusive OR (Selective Set A)	(Y) ∘ (A_A)→A_A	II	Y	Y	2	1.3
01 1	SC	Selective Clear A	(A_A) ∘ (Y)'→A_A	II	Y	Y	2	1.3
01 2	MS	Selective Substitute	(Y)$_n$→(A_{A+1})$_n$ for all (A_A)$_n$=1; (A_A)$_n$=(A_A)$_n$	II	Y	Y	2	1.3
01 3	XOR	Exclusive OR (Sel. Comp. A)	(Y) $\bar{\circ}$ (A_A)→A_A; (A_A)$_n$→(A_A)$_n$ for (Y)$_n$=1	II	Y	Y	2	1.3
01 4	ALP	Add Logical Product	(A_{A+1}) + (Y) ∘ (A_A)→A_{A+1}; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	1.3
01 5	LLP	Load Logical Product	(Y) ∘ (A_A)→A_A	II	Y	Y	2	1.3
01 6	NLP	Subtract Logical Product	(A_{A+1})–(Y) ∘ (A_A)→A_{A+1}; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	1.3
01 7	LLPN	Load Logical Product Next	(Y) ∘ (A_A)→A_{A+1}; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	1.3
02 0	CNT	Count Ones	No. of Bits Set in (Y)→A_A	II	Y	Y	2	11.8
02 2	XR	Execute Remote	(Y)→U	II	N	N	8	1.6
02 3	XRL	Execute Remote Lower	(Y)$_L$→U	II	N	N	8	1.6
02 4	SLP	Store Logical Product	(A_{A+1}) ∘ (A_A)→Y; (A_A)$_i$=(A_A)$_i$t; (A_{A+1})$_i$=(A_{A+1})$_i$t	II	Y	Y	2	2.3†
02 5	SSUM	Store Sum	(A_A)+(A_{A+1})→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.3†
02 6	SDIF	Store Difference	(A_{A+1})–(A_A)→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.3†
02 7	DS	Double Store A	(A_{A+1}, A_A)→Y+1, Y	II	N	N	2	3.4
03 0	ROR	Replace Inclusive OR	(Y) ∘ (A_A)→A_A & Y	II	Y	Y	2	2.6
03 1	RSC	Replace Selective Clear	(A_A) ∘ (Y)'→A_A & Y	II	Y	Y	2	2.6
03 2	RMS	Replace Selective Substitute	(Y)$_n$→(A_{A+1})$_n$ & Y for all (A_A)$_n$=1; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.6
03 3	RXOR	Replace Exclusive OR	(Y) $\bar{\circ}$ (A_A)→A_A & Y; (A_A)$_n$'→A_A & Y for Y$_n$=1	II	N	Y	2	2.6
03 4	RALP	Replace A+Logical Product	(A_{A+1})+(Y) ∘ (A_A)→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.6
03 5	RLP	Replace Logical Product	(Y) ∘ (A_A)→Y & A_{A+1}; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.6
03 6	RNLP	Replace A–Logical Product	(A_{A+1})–(Y) ∘ (A_A)→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	II	Y	Y	2	2.6
03 7	TSF	Test and Set Flag	If (Y)$_{31}$=0, CD Set EQUAL, 1→Y$_{31}$ If (Y)$_{31}$=1, CD Set UNEQUAL.	II	N	Y	8	2.05
05 0	DL	Double Load A	(Y+1, Y)→A_{A+1}, A_A	II	N	N	2	2.5
05 1	DA	Double Add A	(A_{A+1}, A_A)+(Y+1, Y)→A_{A+1}, A_A	II	N	N	2	3.1
05 2	DAN	Double Subtract A	(A_{A+1}, A_A)–(Y+1, Y)→A_{A+1}, A_A	II	N	N	2	3.1
05 3	DC	Double Compare	Compare (A_{A+1}, A_A) to (Y+1, Y), Set CD	II	N	N	2	2.5
05 4	LBMP	Load Base and Memory Protection	(Y)$_{12}$→S_A; (Y+1)$_{20-20}$→SPR$_A$; Y→SIR$_A$ Privileged if; ASR bit 8=0, s ≠ 7, a=7	II	N	N	2	7.7
06 0	FA	Floating-point Add	Shift (A_{A+1}) or (Y+1) Right such that (A_A)–(Y) (A_{A+1})+(Y+1)→A_{A+1}; Normalize	II	N	N	2	7.9
06 1	FAN	Floating-point Subtract	Shift (A_{A+1}) or (Y+1) Right such that (A_A)–(Y) (A_{A+1})–(Y+1)→A_{A+1}; Normalize	II	N	N	2	7.9
06 2	FM	Floating-point Multiply	(A_A)+(Y)→(A_A) (A_{A+1}) • (Y+1)→A_{A+1}; Normalize	II	N	N	2	14.5
06 3	FD	Floating-point Divide	(A_A)–(Y)→(A_A) (A_{A+1})÷(Y+1)→A_{A+1}; Normalize	II	N	N	2	14.5
06 4	FAR	Floating-point Add with Round	Same as FA with (A_{A+1}) rounded	II	N	N	2	9.1
06 5	FANR	Floating-point Subtract w/Rd.	Same as FAN with (A_{A+1}) rounded	II	N	N	2	9.1
06 6	FMR	Floating-point Multiply w/Rd.	Same as FM with (A_{A+1}) rounded	II	N	N	2	14.5
06 7	FDR	Floating-point Divide w/Rd.	Same as FD with (A_{A+1}) rounded	II	N	N	2	14.5
07 0 a=0	XS	Enter Executive State	sy+(B$_b$)→CMR 156; Enter class IV (Executive)	II	N	N	11	5.0†
07 0* a=1	IPI	Interprocessor Interrupt	Send Class II interrupt to processors n (0-1) Specified by (sy+B$_b$)$_n$=1	II	N	N	11	5.0†
07 1**	AEI	Allow Enable Interrupt	Allow Monitor interrupts from IOCa on Channels n; where (sy+B$_b$)$_n$=1	II	N	N	6	3.8†
07 2**	PEI	Prevent Enable Interrupt	Prevent Monitor interrupts from IOCa on channels n; where (sy+B$_b$)$_n$=1	II	N	N	6	3.8†
07 3**	LIM	Load IOC Monitor Clock	sy+(B$_b$)→IOCa MON CLK	II	N	N	6	3.8†
07 4**	IO	Initiate I/O	Initiate IOCa at address Y	II	N	N	2	3.8†
07 5*	IR	Interrupt Return	Return to State Specified by DSW	II	N	N	9	3.75†
07 6	RP	Repeat	Repeat N.I.B) Times; sy+B$_b$(of N.I.)→B$_b$ each cycle	II	N	N	6	1.3
10	LA	Load A	Y→A_A	I	Y	Y	1	1.3
11	LXB	Load A and Index B	Y→A_A; (B$_b$)+1→B$_b$	I	Y	N	1	2.2†
12	LDIF	Load Difference	Y–(A_A)→A_{A+1}; (A_A)$_i$=(A_A)$_i$t	I	Y	Y	1	1.3
13	ANA	Subtract A	(A_A)–Y→A_A	I	Y	Y	1	1.3

REPERTOIRE OF INSTRUCTIONS (CONT.)

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
14	AA	Add A	(A_A)+Y→A_A	I	Y	Y	1	1.3
15	LSUM	Load Sum	(A_A)+Y→A_{A+1}; (A_A)$_i$=(A_A)$_i$t	I	Y	Y	1	1.3
16	LNA	Load Negative	Y'→A_A	I	Y	Y	1	1.3
17	LM	Load Magnitude	|Y|→A_A	I	Y	Y	1	1.3
20	LB	Load B	Y→B$_b$	I	Y	Y	1	2.35
21	AB	Add B	(B$_b$)+Y→B$_b$; B$_b$ zero extended	I	Y	Y	1	2.65
22	ANB	Subtract B	(B$_b$)–Y→B$_b$; B$_b$ zero extended	I	Y	Y	1	2.65
23	SB	Store B	(B$_b$)→Y	I	Y	Y	1	1.7
24	SA	Store A	(A_A)→Y	I	Y	Y	1	1.7
25	SXB	Store A and Index B	(A_A)→Y; (B$_b$)+1→B$_b$	I	Y	N	1	2.2†
26	SNA	Store Negative	(A_A)'→Y	I	Y	Y	1	2.3†
27	SM	Store Magnitude	|(A_A)|→Y	I	Y	Y	1	2.3†
32	BZ	Clear Bit	0→Y$_{sk}$	I	N	Y	3	2.6
33	BS	Set Bit	1→Y$_{sk}$	I	N	Y	3	2.6
34	RA	Replace Add	(A_A)+(Y)→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	I	Y	Y	1	2.6
35	RI	Replace Increment	(Y)+1→A_A & Y	I	Y	Y	1	2.6
36	RAN	Replace Subtract	(Y)–(A_A)→A_{A+1} & Y; (A_A)$_i$=(A_A)$_i$t	I	Y	Y	1	2.6
37	RD	Replace Decrement	(Y)–1→A_A & Y	I	Y	Y	1	2.6
40	M	Multiply A	(A_A) • Y→A_{A+1}, A_A	I	Y	Y	1	12.1
41	D	Divide A	(A_{A+1}, A_A)÷Y→A_A; Remainder→A_{A+1}	I	Y	Y	1	12.1
42	BC	Compare Bit to Zero	If (Y)$_{sk}$=0, CD Set EQUAL If (Y)$_{sk}$=1, CD Set UNEQUAL	I	N	Y	3	1.6
43	CXI	Compare Index Increment	If (B$_b$) ≥ Y, CD Set OUTSIDE, 0→B$_b$ If (B$_b$) < Y, CD Set WITHIN, (B$_b$)+1→B$_b$	I	Y	Y	1	2.95
44	C	Compare	Compare (A_A) to Y, Set the CD	I	Y	Y	1	1.3
45	CL	Compare Limits	If (A_{A+1}) > Y ≥ (A_A), Set CD WITHIN	I	Y	Y	1	1.6
46	CM	Compare Masked	Compare (A_{A+1}) to (A_A) ∘ Y, Set the CD	I	Y	Y	1	1.3
47	CG	Compare Gated	Compare Y–(A_A) to (A_{A+1}), Set the CD	I	Y	Y	1	2.2
50 0	JEP	Jump on Even Parity	If (A_{A+1}) ∘ (A_A) is Even Parity, jump to Y	III	N	N	1	3.55†
50 1	JOP	Jump on Odd Parity	If (A_{A+1}) ∘ (A_A) is Odd Parity, jump to Y	III	N	N	1	3.55†
50 2	DJZ	Jump Double Precision Zero	If (A_{A+1}, A_A)=0, jump to Y	III	N	N	1	3.25†
50 3	DJNZ	Jump Double Precision Not Zero	If (A_{A+1}, A_A)≠0, jump to Y	III	N	N	1	3.25†
51 0	JP	Jump A Positive	If (A_A) ≥ 0, jump to Y	III	N	N	1	3.25†
51 1	JN	Jump A Negative	If (A_A) < 0, jump to Y	III	N	N	1	3.25†
51 2	JZ	Jump A Zero	If (A_A)=0, jump to Y	III	N	N	1	3.25†
51 3	JNZ	Jump A Not Zero	If (A_A) ≠ 0, jump to Y	III	N	N	1	3.25†
52 0	LBJ	Load B and Jump	P → 1→B$_b$, jump to Y	III	N	N	1	2.5†
52 1	JBNZ	Index Jump B	If (B$_b$) ≠ 0, then (B$_b$)–1→B$_b$, jump to Y	III	N	N	1	3.55†
52 2	JS	Jump sy+B	Jump to sy+(B$_b$)	III	N	N	13	2.5†
52 3	JL	Unconditional Jump Lower	Jump to the Lower of Y	III	N	N	12	2.5†
53 0 a=0	JNF	Jump on No Overflow	If OD is not Set, jump to Y; Clear OD	III	N	N	12	2.5†
53 0 a=1	JOF	Jump on Overflow	If OD is Set, jump to Y; Clear OD	III	N	N	12	2.5†
53 1 a=0	JNE	Jump on Not Equal	If CD ≠, jump to Y	III	N	N	12	2.5†
53 1 a=1	JE	Jump on Equal	If CD =, jump to Y	III	N	N	12	2.5†
53 1 a=2	JG	Jump on Greater Than	If CD >, jump to Y	III	N	N	12	2.5†
53 1 a=3	JGE	Jump on Greater Than or Equal	If CD ≥, jump to Y	III	N	N	12	2.5†
53 1 a=4	JLT	Jump on Less Than	If CD <, jump to Y	III	N	N	12	2.5†
53 1 a=5	JLE	Jump on Less Than or Equal	If CD ≤, jump to Y	III	N	N	12	2.5†
53 1 a=6	JNW	Jump Outside Limits	If CD Outside Limits, jump to Y	III	N	N	12	2.5†
53 1 a=7	JW	Jump Within Limits	If CD Within Limits, jump to Y	III	N	N	12	2.5†
53 2	RJ	Return Jump	P+1→Y, jump to Y+1	III	N	N	12	3.3†
53 2	RJC	Return Jump a=1, 2, 3	If switch a is Set, P+1→Y, jump to Y+1	III	N	N	1	3.3†
53 2*	RJSC	Return Jump a=4, 5, 6, 7	If switch a is Set, Stop; P+1→Y; jump to Y+1	III	N	N	1	3.3†
53 3	J	Manual Jump	Jump to Y	III	N	N	12	2.5†
53 3	JC	Manual Jump a=1, 2, 3	If switch a is Set, jump to Y	III	N	N	1	2.5†
53 3*	JSC	Manual Jump a=4, 5, 6, 7	If switch a is Set, Stop; Jump to Y	III	N	N	1	2.55†
54√	LCT	Load CMR Task	(Y)→CMR$_{sk}$	I	N	Y	3	2.35
55*	LCI	Load CMR Interrupt	(Y)→CMR$_{sk+100}$	I	N	Y	3	2.35

REPERTOIRE OF INSTRUCTIONS (CONT.)

Code	Mnemonic	NAME	DESCRIPTION	F	CA	R	UF	Time‡ μs
56√	SCT	Store CMR Task	(CMR$_{sk}$)→Y	I	N	Y	3	2.6†
57*	SCI	Store CMR Interrupt	(CMR$_{sk+100}$)→Y	I	Y	Y	3	2.6†
60√ i=0	HSCT	Store CMR in A	(CMR$_{sk}$)→A_A	IV A	N	N	4	2.35†
60* i=1	HSCI	Store CMR in A	(CMR$_{sk+100}$)→A_A	IV A	N	N	4	2.35†
61√ i=0	HLCT	Load CMR with A	(A_A)→CMR$_{sk}$	IV A	N	N	4	2.35†
61* i=1	HLCI	Load CMR with A	(A_A)→CMR$_{sk+100}$	IV A	N	N	4	2.35†
62	HLC	Shift Left Circularly	(A_A) Left Shifted End Around→A_A	IV B	N	N	10	1.9†
63	HDLC	Shift Left Circularly Double	(A_{A+1}, A_A) Left Shifted End Around→A_{A+1}, A_A	IV B	N	N	10	1.9†
64	HRZ	Shift Right Fill Zeros	(A_A) Right Shifted, Zero Fill→A_A	IV B	N	N	10	1.9†
65	HDRZ	Shift Right Double, Fill Zeros	(A_{A+1}, A_A) Right Shifted, Zero Fill→A_{A+1}, A_A	IV B	N	N	10	1.9†
66	HRS	Shift Right Fill Sign	(A_A) Right Shifted, Sign Fill→A_A	IV B	N	N	10	1.9†
67	HDRS	Shift Right Double, Fill Sign	(A_{A+1}, A_A) Right Shifted Sign Fill→A_{A+1}, A_A	IV B	N	N	10	1.9†
70 0	HSF	Scale Factor	Normalize (A_A) Shift Count→A_B	IV A	N	N	5	3.1†
70 1	HDSF	Double Scale Factor	Normalize (A_{A+1}, A_A) Shift Count→A_B	IV A	N	N	5	3.1†
70 2	HCP	Complement A	(A_A)'→A_A	IV A	N	N	7	1.3†
70 3	HDCP	Double Complement A	(A_{A+1}, A_A)'→A_{A+1}, A_A	IV A	N	N	7	1.3†
71 0	HOR	Logical Sum	(A_A) ∘ (A_B)→A_A; (A_B)$_i$=(A_B)$_i$t	IV A	N	N	5	1.3†
71 1	HA	Sum	(A_A) + (A_B)→A_A	IV A	N	N	5	1.3†
71 2	HAN	Difference	(A_A) – (A_B)→A_A	IV A	N	N	5	1.3†
71 3	HXOR	Logical Difference	(A_A) $\bar{\circ}$ (A_B)→A_A	IV A	N	N	5	1.3†
71 5	HAND	AND	(A_A) ∘ (A_B)→A_A; (A_B)$_i$=(A_B)$_i$t	IV A	N	N	5	1.3†
72***	—	Load IP BITE	(Other Processor CMR)→CMR57	IV A	N	N	—	10.0†
73***	—	Store IP BITE	(CMR 57)→Other Processor CMR	IV A	N	N	—	10.0†
74 0	HM	Multiply Register	(A_A) • (A_B)→A_{A+1}, A_A	IV A	N	N	5	12.1†
74 1	HD	Divide Register	(A_{A+1}, A_A)÷(A_B)→A_A; Remainder→A_{A+1}	IV A	N	N	5	12.1†
74 2	HRT	Square Root	√(A_{A+1}, A_A)→A_B; Residue→A_{B+1}	IV A	N	N	5	21.1†
74 3	HLB	Load B$_B$ with B$_b$	(B$_b$)→B$_B$	IV A	N	N	5	1.75†
74 4	HC	Compare, Register	Compare (A_B) to (A_A), Set CD	IV A	N	N	5	1.3†
74 5	HCL	Compare Limits, Register	If (A_{A+1}) > (A_B) ≥ (A_A), Set CD WITHIN	IV A	N	N	5	1.9†
74 6	HCM	Compare Masked, Register	Compare (A_{A+1}) ∘ (A_A) to (A_B), Set the CD	IV A	N	N	5	1.3†
74 7	HCB	Compare B$_b$ with B$_B$	Compare (B$_b$) to (B$_B$), Set the CD	IV A	N	N	5	2.05†
77 0**	HSIM	Store IOC Monitor Clock in A	(IOC$_B$ MON CLK)→A_b	IV A	N	N	5	3.8†
77 1	HSTC	Store Real-Time Clock in A	(IOC$_B$ RTC)→A_b	IV A	N	N	5	3.8†
77 4*	HPI	Prevent Class III Interrupts	Set Class III Interrupt Lockout	IV A	N	N	9	1.3†
77 5*	HAI	Allow Class III Interrupts	Clear Class III Interrupt Lockout	IV A	N	N	9	1.3†
77 6*i=0	HALT	Stop Processor	Stop CPU	IV A	N	N	9	2.5†
77 6*i=1	HWFI	Wait for Interrupt	Cease Memory References until Interrupted	IV A	N	N	9	2.5†
77 6***i=1 a=1	—	Allow IP BITE	Allow IP BITE until Interrupted	IV A	N	N	—	2.5†

ULTRA/32 PSEUDO INSTRUCTIONS

10	ZA	Clear A	0→A_A	I	N	Y	7	1.3
20	ZB	Clear B	0→B$_b$	I	N	Y	7	2.35
20	NOOP	No Operation	0→B$_0$	I	N	Y	9	2.35
23	SZ	Store Zero	0→Y	I	Y	Y	12	1.7
74 3	HNO	Half Word No Operation	(B$_0$)→B$_0$	IV A	N	N	9	1.75†

ULTRA/32 FORMATING MNEMONICS

—	HK	Half Word Constant (Variable field becomes next halfword)	—	—	16	—
—	IW	Indirect Word (c=10)	—	—	8	—
—	IWS	Indirect Word, Special Base (c=00, c$_1$=0)	—	—	11	—
—	IWB	Indirect Word, Special Index (c=00, c$_1$=1)	—	—	11	—
—	IWC	Indirect Word, Character (c=01)	—	—	14	—
—	IWCI	Indirect Word, Character Increment (c=11)	—	—	14	—
—	MP	Memory Protection (see SPR format)	—	—	15	—

ULTRA/32 CODING FORMATS (UF)

(An Asterisk (*) Preceding y Indicates Indirect Addressing)

No.	Variable Field	No.	Variable Field	No.	Variable Field	No.	Variable Field	No.	Variable Field
1	a, y, k, b, s	4	af$_4$, b	7	a	10	a, m (shift by m)	11	sy, b	14	y, w, p, b, s
2	a, y, b, s	5	a, b	8	y, b, s		a, b, 1 (shift by B$_b$)	12	y, k, b, s	15	r, i, or, ow, ia, ir
3	ak, y, b, s	6	a, sy, b	9	None		a, b, 2 (shift by A$_b$)	13	sy, k, b	16	e

*Privileged **CPU→IOC Instr.–Privileged √Privileged when ak=2X, 5X, 6X or 7X ***Machine Language Only—Privileged
†Execution time independent of overlap operation
‡Times shown assume 750 ns memory with operands not in same 32K unit as instructions (overlapped).

I/O CONTROLLER COMMANDS

Code	Mnemonic	NAME	DESCRIPTION	UF**	Time μs
01	KECM	Load Control Memory	$(Y) \rightarrow CMR_{ulj}$	1	4
02	KSCM	Store Control Memory	$(CMR_{ulj}) \rightarrow Y$	1	4
03	KSB	Set Bit I_j	$I \rightarrow Y_{Ij}$	2	6
04	KCB	Clear Bit I_j	$0 \rightarrow Y_{Ij}$	2	6
05	KTB	Test Bit I_j	Sense Y_{Ij}; skip if $(Y_{Ij})=(u_{24})$	3	4
06	KTSB	Test and Set Bit I_j	Sense Y_{Ij}; skip if $(Y_{Ij})=0$; Set Y_{Ij}	2	6
07	KJ	Jump to Y	$Y \rightarrow$ Program Address Counter	4	2
10	KIB	Initiate Input Buffer on C_{Ij}	$(Y) \rightarrow CMR$; (u)=Unit; (Ij)=Channel	5	4
11	KOB	Initiate Output Buffer on C_{Ij}	$(Y) \rightarrow CMR$; (u)=Unit; (Ij)=Channel	5	4
12	KEFB	Initiate External Function Buffer on C_{Ij}	$(Y) \rightarrow CMR$; (u)=Unit; (Ij)=Channel	5	4
13	KEIB	Initiate External Interrupt Buffer on C_{Ij}	$(Y_{18\text{-}0}) \rightarrow CMR$; (SMR)=Units; (Ij)=Channel	6	4
14	KIBF	Initiate Input Buffer on C_{Ij} with Force	$(Y) \rightarrow CMR$; (j)=Channel	7	4
15	KOBF	Initiate Output Buffer on C_{Ij}; with Force	$(Y) \rightarrow CMR$; (u)=Unit; (Ij)=Channel	5	4
16	KFBF	Initiate External Function Buffer on C_{Ij} with Force	$(Y) \rightarrow CMR$; (u)=Unit; (Ij)=Channel	5	4
17	KTDB	Terminate Data Buffer on C_j	(j)=Channel; terminate Data Buffer	8	2
20	KTEI	Terminate External Interrupt Buffer on C_j	(j)=Channel; terminate EI Buffer	8	2
21	KI	Initiate Interrupt on C_j	Interrupt CP; $(u_{Ij}) \rightarrow ISC$	5	2
22	KSBX	Store BITE on C_j	(BITE $C_j) \rightarrow Y$; (j)=Channel	7	4
23	KEDB	Enable/Disable BITE on CY	(u)=1-enable; (u)=0-disable; (Y)=Channels	9	4
24	KESM	Load External Interrupt Scan Mask on C_j	$(Y_{7\text{-}0}) \rightarrow$ Scan Mask; (j)=Channel	7	4
25	KSSM	Store External Interrupt Scan Mask on C_j	(Scan Mask $C_j) \rightarrow Y$	7	4
26	KSBS	Store Buffer Status on C_j	$(BCW_{j31\text{-}0}) \rightarrow Y_{+1}$; $(BCW_{j59\text{-}32}) \rightarrow Y$	7	6
27	KRB	Resume Buffer on C_{Ij}	$(Y) \rightarrow BCW_{j59\text{-}32}$; $(Y_{+1}) \rightarrow BCW_{j31\text{-}0}$	6	6
30	KRCH	Release Channel Hold on C_j	(j)=Channel	8	2
31	KRTC	Load Real Time Clock	$(Y_{31\text{-}16}) \rightarrow RTC_{31\text{-}16}$; $177777 \rightarrow RTC_{15\text{-}0}$	4	4
		FORMATING MNEMONICS			
—	KBCW	Buffer Control Word		10	—

CMR—Control Memory Register
ISC—Interrupt Status Code
SCM—Scan Mask Register

**ULTRA FORMAT

1—ul_j,y, c	4—y, c	7—y, j, c	10—y, l, m (l=buffer length)
2—l_j, y, c	5—l, y, j, u, c	8—j, c	
3—l_j, y, u, c	6—l, y, j, c	9—y, u, c	(m=monitor)

DATA BUFFER CONTROL WORD FORMAT

59	58	57	56	55 54	53 51	50	49 32	31	30 18	17 0
Data Chain Request Active CR	Data Buffer Active BA	IOI Identifier I	Force F	Buffer Type r: 00—In, 01—Out, 10—EF, 11—Not Used	Unit Code	Chain Flag c	Command Pointer Address	Monitor Flag m	Final Buffer Address	Initial Buffer Address

IOC COMMAND WORD FORMAT

31 27	26 24	23	22 19	18	17 0
Function Code f	Unit Code Designator u (for selected Channel)	IOI Designator I	IOI Channel Designator j	Chain Flag c	Operand or Operand Address y

EI BUFFER CONTROL WORD FORMAT

18	17 0
Monitor Flag m	EI Buffer Address

CHAIN BASE CONTROL WORD FORMAT

17 6	5	4 2	1 0
Chain Base Address	IOI Identifier I	Unit Code	Request Type R: 00—"101" Request, 01—"101" Storage Request, 10—Error Request, 11—Not Used

IOC CONTROL MEMORY ADDRESS ASSIGNMENT

IOC CMR Address	Register Group	Register Size
0-17	Data Buffer Control Group (Bits 31-0 of Data BCW)	32 Bits
20-37	Data Buffer Control Group (Bits 59-32 of Data BCW)	28 Bits
40-57	External Interrupt Buffer Control Group	19 Bits
60-77	Chain Base Control Group	12 Bits

INTERRUPT STATUS CODES

Class	INTERRUPT	Status Code Bits** 12 11 10 9 8 7 6 5 4 3 2 1 0
I*	Power Tolerance (Never Locked Out)	0 0 0 0 0 0 1 0 0 0
I	CP—Operand Memory Resume	0 0 M M M M 0 0 0 0
I	CP—IOC Command Resume	K 0 0 0 0 0 0 0 0 1
I	CP—Instruction Memory Resume	0 0 M M M M 0 0 1 0
I	CP—IOC Interrupt Code Resume	K 0 0 0 0 0 0 0 1 1
I*	Memory Parity Error	0 0 M M M M 1 0 0 1
I*	IOC Memory Resume	K 0 M M M M 1 0 1 0
I*	IOC Illegal CAR Command/Instruction	K 0 0 0 P W 1 0 1 1
I*	IOC Illegal Chain Command	K T C C C C 1 1 F F
II*	Interprocessor Interrupt	0 0 0 0
II	Floating Point Error	0 0 0 1
II	CP Illegal Instruction Error	0 0 1 0
II	Privileged Instruction Error	0 0 1 1
	Not Assigned	0 1 0 0
II	Operand Breakpoint Match	0 1 0 1
II	Operand Read or Indirect Addressing	0 1 1 0
	Not Assigned	0 1 1 1
	Not Assigned	1 0 0 0
II	Operand Write	1 0 0 1
II	Operand Limit	1 0 1 0
II	Instruction Breakpoint Match	1 0 1 1
	Not Assigned	1 1 0 0
II	Instruction Execute	1 1 0 1
II	Instruction Limit	1 1 1 0
II*	CP Monitor Clock	1 1 1 1
III*	IOC Monitor Clock	0 0 0 K 0 0 0 0 0 1 0 0 1
III*	IOC Illegal Buffer Initation	0 0 0 K T C C C C 1 0 1 0
III*	IOC—CP Interrupt	U U U K T C C C C 1 0 1 1
III*	IOC External Interrupt Monitor	U U U K T C C C C 1 1 0 0
III*	IOC External Function Monitor	0 0 0 K T C C C C 1 1 0 1
III*	IOC Output Data Monitor	0 0 0 K T C C C C 1 1 1 0
III*	IOC Input Data Monitor	0 0 0 K T C C C C 1 1 1 1
IV	Executive Return	16 bit code assigned thru program

*Queued

**Definitions: MMMM—16K Memory Banks (0-17)
CCCC—IOI Channel (0-17)
UUU—Peripheral Unit (0-7)
W-0-Illegal IOC Command
I-Illegal CP-IOC Instruction
P—CP (0, 1)
K—IOC (0, 1)
T—IOI (0, 1)
FF—Function
00—Data Chain
01—Error Chain
10—"IOI" Chain
11—Not Used

MEMORY PROTECTION REGISTERS

Storage Protection Register (SPR)

20	19	18	17	16	15 0
I	OR	OW	IA	IR	R
Allow Instruction Execution*	Allow Operand Reading*	Allow Operand Writing*	Allow Indirect Addressing*	Allow use of Interrupt Index and Base Registers*	Displacement Value

*Operation Allowed if Bit is Set

SEGMENT IDENTIFICATION REGISTER (SIR)

20	19 17	16	15 0
	SIR_s		SIR_d
	Base Register Designator		16 Bit Displacement

BREAKPOINT REGISTER

19	18	17 Comparison Address Bits 0
0	0	—Disabled
0	1	—Instruction address
1	0	—Operand address
1	1	—Instruction and operand addresses

CENTRAL PROCESSOR CONTROL MEMORY ADDRESS ASSIGNMENT

Address	Use	Bits
	Task Mode	
0-7	Accumulator (A) registers 0-7	32
10	Unassigned	19
11-17	Index (B) registers 1-7	19†
20-27	Base (S) registers 0-7**	18
30-47	Unassigned (not usable)	—
	Interrupt Mode	
50	Power Status Register**	32
51	Functional Status Register**	32
52-56	Unassigned (not usable)	—
57	Interprocessor BITE Storage**	32
6x	Breakpoint register**	20
7x	Active status register**	21
100-107	Accumulator (A) registers 0-7	32
110	CP monitor clock register	19*
111-117	Index (B) registers 1-7	19†
120-127	Base (S) registers 0-7	18
130-137	Unassigned (not usable)	—
140	ICW—Class I	20
141	DSW—Class I ASR storage	20
142	DSW—Class I interrupt status code	20
143	DSW—Class I P—storage	20
144	ICW—Class II	20
145	DSW—Class II ASR storage	20
146	DSW—Class II interrupt status code	20
147	DSW—Class II P—storage	20
150	ICW—Class III	20
151	DSW—Class III ASR storage	20
152	DSW—Class III interrupt status code	20
153	DSW—Class III P—storage	20
154	ICW—Class IV	20
155	DSW—Class IV ASR storage	20
156	DSW—Class IV interrupt status code	20
157	DSW—Class IV P—storage	20
160-167	Storage Protection Registers (SPR) 0-7	21
170-177	Segment Identification Registers (SIR) 0-7	21

*Clock is 16 Bits

**Not Addressable in the Task Mode. (Privileged instruction error will occur)

†Lower 16 bits used for index and arithmetic functions. Upper three bits used **only** as a base-register designation.

ACTIVE STATUS REGISTER

Bit	Designator
20	Central Processor Identifier—Hardwired
19	State I (Hardware Control)
18	State II (Hardware Control)
17	State III (Hardware Control)
16	State IV (Hardware Control)
15	Upper—lower (Hardware Control)
14	Class I lockout
13	Class II lockout
12	Class III lockout
11	Base (S) register selector
10	Accumulator/B register selector
9	Memory lockout inhibit
8	Load base enable
7	Bootstrap mode
6-4	Usable spare bits
3	Fixed point overflow indicator
2	0—Not equal 1—Equal
1	0—Less than 1—G.T. or equal
0	0—Within limits 1—Outside limits

Bits 9-11 1—Interrupt mode
0—Task mode

FLOATING POINT FORMAT (each word is one's complement)

Sign Fill ±	14 0	±	30 0
Characteristic (exponent) in A_a or Y		Mantissa in A_{a+1} or Y+1	

INSTRUCTION WORD FORMATS

Format I

31 26	25 23	22 20	19 17	16	15 13	12 0
f	a	k	b	i	s	y

Format II

31 26	25 23	22 20	19 17	16	15 13	12 0
f	a	f_2	b	i	s	y

Format III

31 26	25 23	22 21	20	19 17	16	15 13	12 0
f	a	f_3	x	b	i	s	y

Format IV A

31 26 / 15 10	25 23 / 9 7	22 20 / 6 4	19 17 / 3 1	16 / 0
f	a	f_4	b	i

Format IV B

31 26 / 15 10	25 23 / 9 7	22 16 / 6 0
f	a	m

NORMAL INDIRECT ADDRESS WORD FORMAT

31 30	29 25	24 20	19 17	16	15 13	12 0
c	w	p	b	i	s	y

SPECIAL INDIRECT ADDRESS WORD FORMAT

31 30	29	28 20	19 17	16	15 0
c	c_1	x	b	i	d

f—Function Code
f_2 f_3 f_4—Subfunction Codes
a—Accumulator Register
k—Operand Interpretation
b—Index Register
i—Indirect Bit
s—Base Register
w—Field Width
p—Bit Position
y—Operand Address
x—Not Used
d—16 Bit Displacement

c_1—Special Indirect Subfunction
0—$Y=d+(S_b)$
1—$Y=d+(B_b)+(S)$ as specified by $(B_b)_{19\text{-}17}$
c—Addressing Designator
00—Indirect Special
10—Indirect Normal
01—Single Character
11—Sequential Character

m—Shift Designator

Bit 26	Function
0	Shift by count 2^5—2^0
1	Shift by B_b if $2^5=0$
1	Shift by A_b if $2^5=1$
b is specified by bits 2^3—2^1	

REPEAT CONDITIONS

a	Non-Compare Instructions
0	Terminate if $A \neq 0$
1	Terminate if $A = 0$
2	Terminate if $A \geq 0$
3	Terminate if $A < 0$
4	Do not terminate
5	Terminate if (A) is even parity on write into memory
6	Terminate if (A) is odd parity on write into memory
7	Do not terminate

a	Compare Instructions
0	Terminate if CD set to $\neq$
1	Terminate if CD set to $=$
2	Terminate if CD set to $>$
3	Terminate if CD set to $\geq$
4	Terminate if CD set to $<$
5	Terminate if CD set to $\leq$
6	Terminate if CD set to outside limit
7	Terminate if CD set to within limit

FORMAT I INSTRUCTION k—FIELD INTERPRETATION

k	Memory to Arithmetic (Read)	Arithmetic to Memory (Store)
0	$y_{SE}+(B_b) \rightarrow A_{15\text{-}0}$ SE	Not I[illegible]
1	$(Y_{15\text{-}0}) \rightarrow A_{15\text{-}0}$ SE	$(A_{15\text{-}0}) \rightarrow Y_{15\text{-}0}$; $Y_{31\text{-}16}$—Un
2	$(Y_{31\text{-}16}) \rightarrow A_{15\text{-}0}$ SE	$(A_{15\text{-}0}) \rightarrow Y_{31\text{-}16}$; $Y_{15\text{-}0}$—Un
3	$(Y_{31\text{-}0}) \rightarrow A_{31\text{-}0}$	$(A_{31\text{-}0}) \rightarrow Y_{31\text{-}0}$
4	$(Y_{7\text{-}0}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{7\text{-}0}$; $Y_{31\text{-}8}$—Un
5	$(Y_{15\text{-}8}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{15\text{-}8}$; $Y_{31\text{-}16}$—Un $Y_{7\text{-}0}$—Un
6	$(Y_{23\text{-}16}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{23\text{-}16}$; $Y_{31\text{-}24}$—Un $Y_{15\text{-}0}$—Un
7	$(Y_{31\text{-}24}) \rightarrow A_{7\text{-}0}$ ZE	$(A_{7\text{-}0}) \rightarrow Y_{31\text{-}24}$; $Y_{23\text{-}0}$—Un

k—Field Interpretation for Replace Instructions:
Read Cycle—Same as memory to arithmetic.
Store Cycle—Same as arithmetic to memory. For Repeat, with b of repeat instruction not zero, Y will be modified by S_6 and not S_s for store cycle.

SE—Sign Extended; ZE—Zero Extended; Un—Unchanged

SYMBOL DEFINITIONS

CMR—Control Memory Register
F—Format
CA—Character Addressable
R—Repeatable
DSW—Designator Storage Word
UF—Ultra Format
$(A)_n$—Contents of A, bit n
CD—Compare Designator
Y—Address formed by $y+(B_b)+(S_s)$
ICW—Initial Condition Word
OD—Overflow Designator
$\underline{Y}$—Operand (Y) (Whole word or partial word) or Y, depending on k
$\odot$—Logical product (AND)
$\oplus$—Logical sum (Inclusive OR)
$\overline{\oplus}$—Logical difference (Exclusive OR)